TOWARDS AN AGENT-BASED, AUTONOMOUS TACTICAL SYSTEM FOR C4ISR OPERATIONS

R. J. Keisler
SPAWAR Systems Center Atlantic
N. Charleston, SC 29419

ABSTRACT

Software agent involvement in combat computer systems can provide lightweight, dynamic automation to the warfighter. Current intelligence systems are limited by the abilities of the warfighters to gather, assimilate, report, and execute on intelligence. Through the research and design for the Overshadow system, we found using agents practical for automation capabilities in tactical systems where optimization of operation and human simulation are important factors. In the paper we provide an overview of the architecture and design decisions made during the Overshadow system development.

INTRODUCTION

For tactical operations, every second counts. Up-to-date intelligence and commands are crucial not only for successful mission completion but to save soldier's lives. To accomplish many of these real-time tasks, warfighters must rely on support from body worn or vehicularized computer systems for command and control, communications, computers and intelligence, surveillance, and reconnaissance (C4ISR)—systems that must work within new net-centric battlespaces.

Distributed Common Ground System (DCGS) is a major effort in architecting the future for combat systems in these net-centric battlespaces. DCGS is the integration architecture for all ground/surface systems with respect to information collection, processing, and exploitation.

DCGS represents a major change from past system design methodologies, correctly making information sharing and joint operations visible to the warfighter. To support these goals, new system designs must be dynamic, distributed, scalable, and sociable—able to join heterogeneous communities without disruption. No longer will resources, databases, and sensors be limited by the visibility and control of small set of users, but they will be exposed to many users with disparate goals and concerns.

To accomplish proper collaboration within these new information societies, intelligence systems can no longer fully rely on human operators for manual control of all the resources, sensors and systems. As these societies grow, human efficiency will not be adequate to execute data processes like collection, analysis, and dissemination. Independent, assistive entities known as agents must be trusted to solve this problem and to coexist with human operators.

1. AGENTS

An agent is a software construct that is capable of flexible autonomous action in dynamic, unpredictable, and social environments (Huhns, 2005). Usually agents are small in comparison to today's manual, monolithic computer systems this is because an agent's power is derived from a social arrangement in a multi-agent system. Agents are a disruptive, leading edge technology within computing, both well understood and highly researched, and serve a purpose unfulfilled by other solutions. Their fundamental use is becoming more necessary as combat systems become more open, distributed, and scaled.

The use of agents is also driven by a few major technology factors (AgentLink III, 2005; Hendler, 2001; Huhns, 2005). First, the data new systems create and analyze is becoming more descriptive through the use of semantic ontologies, which layer the data with common meaning and understanding. Next, computing systems are moving towards discoverable, actionable service interfaces to standardize the interaction with sensors, resources, and systems spread throughout dynamic networks. Finally, automation mechanisms are required to collaborate with and control these ever increasing sets of data, services, and sensors as systems move to scales no longer amenable to human control. Agents are uniquely placed to autonomously manage such factors, and in fact were created with these premises in mind.

Through communication and sharing, agents can combine their knowledge into dynamic representations of the environment. Also, by incorporat-

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **01 DEC 2008**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE
Towards An Agent-Based, Autonomous Tactical System For C4isr Operations

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
SPAWAR Systems Center Atlantic N. Charleston, SC 29419

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR’S ACRONYM(S)

11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM002187. Proceedings of the Army Science Conference (26th) Held in Orlando, Florida on 1-4 December 2008

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**UU**	**6**	

ing learning, simulation, and programmed intelligence, agents can help with decision and action processes by offering what they understand as the most beneficial options. These pictures and suggestions offered in real-time will allow warfighters at tactical levels to make more informed decisions.

2. OVERSHADOW

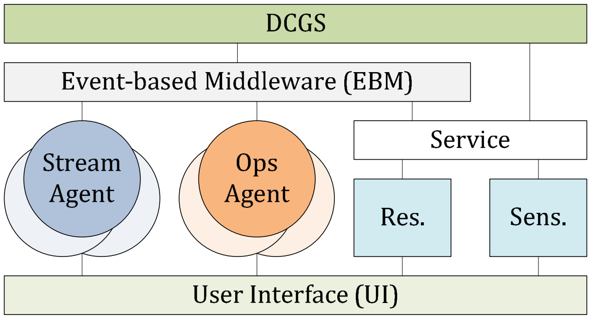

Fig. 1: Overshadow architecture

Overshadow (Figure 1) is a data-centric, agent-based system. Its design is driven by the need for automated support for human operators in tactical intelligence, surveillance, and reconnaissance (ISR) systems. While the primary focus is on ISR, the design can also translate into systems for Command and Control (C2) or Command and Control, Communications, and Intelligence (C3I) systems.

Data-centricity allows Overshadow to capture ISR processes—such as collection, analysis, and dissemination—within its system scope and also enables an operator to insert his or her presence within any of the processes. The data also allows for Overshadow to provide specific data type integration with DCGS. On top of the data, agents afford a flexible, sociable automation system, supporting process and service control. To adapt the agents for use in a tactical system quality of service (QoS) constraints needed to be used.

The QoS constraints are to support best-case performance during tactical operation and restrict agent operation within a set of general requirements:

- Intelligence generation must be timely and prioritized.
- Execution elements must operate efficiently.
- The system must be unobtrusive to the operator.

These requirements ensure the agents do not take vital time and attention away from the warfighter, limiting their focus when under-fire or in fast-paced missions.

The design process for Overshadow merged agent requirements and other important concepts within information architectures. To support the design's data-up policy the first step was defining the basic structure of the data. Next, the resources and sensors of the system were hidden behind service interfaces and they leveraged an event-based network for communication and coordination. Finally, agents were constructed for data fusion and service control.

2.1 Data

The major backbone of any information system is data, because that data feeds the processes that generate actionable information (in the case of ISR systems information is intelligence). By examining common requirements on data in commercial systems, a design decision was made to require all of the data to define its priority, the location it was generated at and its provenance. Other meta-properties are reserved for the implementations and the standards they follow, and a data semantics standard such as the Resource Definition Format (RDF) or the Web Ontology Language (OWL) would be useful.

The reason the Overshadow system data must tag an initial location is it allows the system to build location-optimized communities of interest within the networks. These communities are beneficial as the messages that carry the data should not be distributed to places where they are not useful, wasting bandwidth in the process. The location information is also useful when querying sets of data generated in an area of interest.

Provenance is a relatively new requirement within data processing, but is very useful for reporting purposes. It is a common means to track physical objects that processed data, from sensors and resources to agents and people. Overshadow uses unique identifiers to tag resources—generated at runtime or provided by manufacturers. This is useful within the system because of how these objects are hidden from direct access.

2.2 Resources and Sensors

As shown in Figure 2 resources and sensors are hidden behind service interfaces. These interfaces must have definitions for three meta-properties: unique identification, location, and semantic inputs and outputs (I/O). The service meta-properties correlate to the data meta-properties required for all the data Overshadow manages, and allow its agents to use the properties for internal tasking. The interfaces may also have definitions in the Web Services Definition Language (WSDL) for use by other systems.

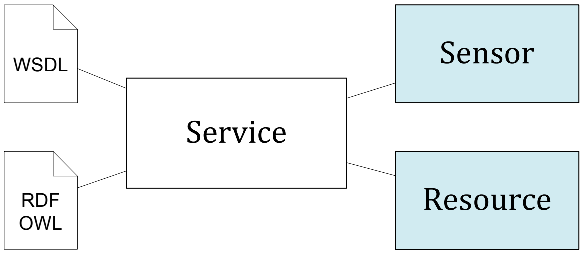

Fig. 2: Service interfacing to resources and sensors

To integrate with DCGS, a service defined by a WSDL can register itself with the UDDI registry just like any other web service. There are also plans to supply abilities for a service to connect directly to the DCGS Integrated Backbone (DIB) for use in data fusion beyond the system scope. Then, as the service receives requests, the result data will be sent back to the requestors and a copy of the metadata put into the DIB. The data-binding with the DIB will be based on priority or user input; only important data needs to push beyond Overshadow.

Beneath the scope of DCGS, Overshadow is designed so that its services may be randomly distributed throughout the network. However, realistically there will be clustering patterns aligned to system or location. There are many approaches to manage this distribution; a few explored during the system design are:

- Use a single high-level registry to find services (ex: UDDI).
- Deploy a peer-to-peer (P2P) network of registries.
- Engage with P2P event-based middleware (EBM).

The P2P-EBM approach was finally chosen primarily because it eliminates discovery and some of the composition from the agents, which are NP-complete problems (Schlegel, 2007; Stein, 2007), by using event subscriptions.

Most major commercial-off-the-shelf (COTS) message-oriented middleware (MOM) solutions and standards such as Extensible Messaging and Presence Protocol (XMPP) will have basic support for event-based operation, but do not necessarily come with network optimization. The first generation of Overshadow will rely on a JMS broker network connected via P2P mechanism as its initial event network. Future generations of the system will benefit from work ongoing to create a robust epidemic multicast network for event distribution (Costa, 2004; Kermerrac, 2003). However, as long as there is some abstraction between the software and the networks, any event-based middleware can be used.

When using the EBM for communications and control it was known that normal techniques for service and process management would not work. Most of these schemes rely on direct access to the service, but the EBM provides one publish-subscribe channel for all the data. Sending directed, specialized events could cause unnecessary complexities in optimization algorithms and would negate the reason for choosing the EBM over a registry-based solution. The solution to this problem was using naturally distributed and communicative agents to execute the service and process control.

3. OVERSHADOW AGENTS

In designing Overshadow, many options for providing automation capabilities to tactical computer systems were explored. Software agents seemed the best fit, but due to community skepticism for agent use in current computer systems they were not the first choice.

Initially, a trade study was conducted over many COTS based solutions that could aid in automating the interaction between services, including bussing (ESB), orchestration (WS-BPEL, XPDL), choreography (WS-CDL), and other non-standard products. The final determination was that those solutions would have too much communications and processing overhead, would be too hard to control, and have little flexibility all necessary qualities for a tactical system. In addition, the solutions imposed too many requirements on higher-level systems that were outside of Overshadow's scope of control.

Leveraging software agents showed overwhelming benefits that overcame the costs associated with their use. The agents required little to no configuration, could be constrained in their execution and communication, and were able to mimic human functions better than any other choices. Also, as human-computer interaction (HCI) was a major concern, and always is within automation systems (Parasuraman, 2000; Parasuraman, et al., 2000; Shneiderman, 2007), Overshadow's simplistic agents can act as a more realistic bridge between operators and the information management process.

As for agents in Overshadow, there are two major patterns. The Stream Agent is the information manager, and the Ops Agent is the virtual system operator.

3.1 Stream Agents

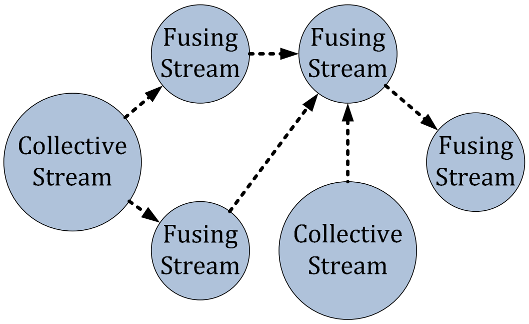

Fig. 3: Stream Agents

The first and more important agent pattern in the Overshadow system is the Stream Agent or Streams (Figure 3). Streams are the suppliers and managers of vertical data, distributing it with a single notification type through the EBM. They are tasked with pulling all of the resources and sensors within a system together into realistic data constructs. The agents come in two sub-patterns: Fusing and Collective.

3.1.1 Fusing Streams

The Fusing Stream (Figure 4), is very loosely based on the Joint Directors of Laboratories (JDL) data fusion model (Llinas, 2004). In accordance with the model, the agent's purpose is to create and refine the process for acquisition and fusion of its data. The Fusing Stream requires a semantic class definition taken from the data semantics, which describes all of the data it must fuse (Hendler, 2001; Pu, 2006). The agent uses this definition to create an internal workflow and to bind to the appropriate events within the EBM.

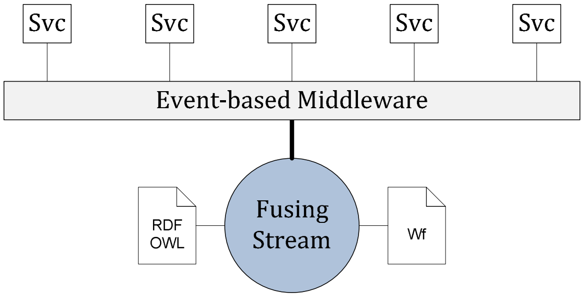

Fig. 4: Fusing Stream configuration

The Fusing Stream's internal workflow is based on orchestration concepts and managed by a very simplistic workflow engine. A standard for orchestration and workflow description such as Business Process Execution Language (BPEL) or a COTS workflow engine could have been used, but those choices would create size and configuration complexity the agent did not need. The simplistic, adaptive workflow engine the agents utilize is more suited for their applications. In the future the agent will sense its level of input starvation and internal congestion, and use the statistics for internal process optimization, input filtering, load balancing, and other helpful functions for a tactical system.

3.1.2 Collective Streams

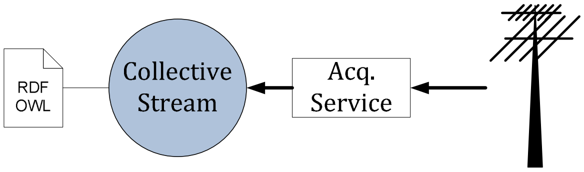

Fig. 5: Collective Stream acquisition process

The Collective Stream (Figure 5) is the simplest implementation of a Stream Agent and is the beginning of Overshadow's collection management process. The Stream itself is bound to an acquisition service, utilizing its service to collect tips or initial data sets. This data will be used as a basis for more descriptive analysis by other agents or operators. Some examples of Collective Streams are a receiver scanning for targets, a service periodically querying a database, or sensors monitoring environmental properties. The future of this agent lies in the ability to add artificial intelligence for predictive analysis that can auto-manage the service without outside interaction.

3.2 Ops Agents

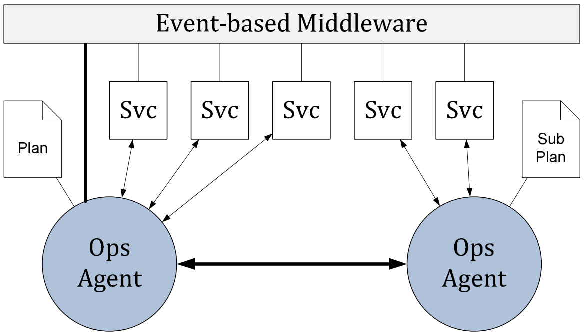

Fig. 6: Ops Agent configuration

The second agent pattern Overshadow defines, the Ops Agent (Figure 6), is the virtual operator for the system. The agent works with a user-defined (or even higher-level agent-defined) plan with rules and goals that governs the adaptation of resources, sensors, and agents a system owns to the current operation environment.

As shown in Figure 6, Ops Agents can be tiered in their command structure, but each system should own at least one to monitor its personal resources. However, the agent does not need to be restricted to one-to-many interaction, and some implementations may find that a few one-to-one/agent-to-service interactions are required.

The internals of Ops Agents can be varied depending on system complexity. For initial versions the agents will be relatively unintelligent following simple plans, but later versions could incorporate artificial intelligence such as Bayesian analysis, Markov models, genetic algorithms, etc. However, Overshadow requires the agents support some adjustable autonomy, such as on, off, or guidance states depending on operator presence or configuration for HCI purposes.

4. FUTURE WORK

There are many open software frameworks for agent development. The Java Agent Development Framework (JADE) was initially selected for the basis of Overshadow's agent architecture. However, while agent construction was easy within the framework, interacting with the agents from user interfaces was not. Further, using XML-based standards for communication within tactical systems is not efficient even with compression. Therefore, it was acknowledged that there is a need for further research into agent frameworks and lightweight communication protocols for use in tactical systems. Standardization of these elements is necessary for future robust, net-centric agent systems. The design of Overshadow tried to address these problems, but a standard was beyond the scope of the requirements.

CONCLUSION

Agent involvement in combat computer systems can become the next revolution in automation and human-computer interaction. While there are many doubters, agent technology is ready now if new concepts in computer systems are adopted. Through the research and design for Overshadow, we found an agent-based architecture a viable alternative to other service-oriented architectures (SOA) for tactical systems where optimization of operation and human simulation are more important.

The future warfighter will benefit from systems harnessing the power of agents, because agents can provide force multiplication without the need for more boots on the ground. Agents will not only exist as collaboration and information providers, but learning, adaptive virtual soldiers and commanders. Agents are just now beginning to disrupt the old models for computer systems, and should continue to be a choice in future system designs.

ACKNOWLEDGEMENTS

I would like to thank Mr. Thomas Kirkpatrick, Systems Engineer for the Information Warfare Division of SPAWAR Systems Center Atlantic (SSC-LANT) for the many discussions of current and future C4ISR systems and technology. Also, I would like to thank Mr. Glenn Segelken of SSC-LANT and

Dr. Paul Buhler of the College of Charleston for their time discussing SOA, agents, events and much more within both the academic and military communities. Finally, I would like to thank Dr. Suzanne Huerth of SSC-LANT for her expert guidance and review. This work was supported by funding from SSC-LANT FY08 Innovation and the Joint Threat Warning System (JTWS) SMART R&D Effort.

REFERENCES

AgentLink III, 2005: Agent Technology: Computing as Interaction: A Roadmap for Agent Based Computing. [Available online at http://www.agentlink.org/roadmap/al3rm.pdp.]

Costa, P., M. Migliavacca, G. P. Picco, G. Cugola, 2004: Epidemic Algorithms for Reliable Content-Based Publish-Subscribe: An Evaluation. Proc. 24th International Joint Conf. on Distributed Computing Systems, Tokyo, Japan, ICDCS.

Hendler, J., 2001: Agents and the Semantic Web. IEEE Intelligent Systems, 16:2, 30-37.

Huhns, M. N., et al., 2005: Research Directions for Service-Oriented Multiagent Systems. IEEE Internet Computing, 9:6, 65-70.

Kermarrec, A., L. Massoulié, and A. J. Ganesh, 2003: Probabilistic Reliable Dissemination in Large-Scale Systems. IEEE Trans. on Parallel and Distributed Systems, 14:3, 248-258.

Llinas, J., C. Bowman, G. Rogova, A. Steinberg, and F. White, 2004: Revisiting the JDL data fusion model II. Proc. 7th Conf. on Information Fusion, Stockholm, Sweden, ISIF, 1218-1230.

Parasuraman, R., 2000: Designing automation for human use: empirical studies and quantitative models. Ergonomics, 43:7, 931-951.

Parasuraman, R., T. B. Sheridan, and C. D. Wickens, 2000: A Model for Types and Levels of Human Interaction with Automation. IEEE Trans. on Systems, Man, and Cybernetics--Pt. A: Systems and Humans, 30:3, 286-297.

Pu, K., V. Hristidis, and N. Koudas, 2006: A Syntactic Rule Based Approach to Web Service Composition. Proc. 22nd Conf. on Data Engineering, Atlanta, GA, IEEE Computer Society, 31.

Schlegel, T. and R. Kowalczyk, 2007: Towards Self-organizing Agent-based Resource Allocation in a Multi-Server Environment. Proc. 6th International Joint Conf. on Autonomous Agents and Multiagent Systems, Honolulu, HI, IFAAMAS.

Shneiderman, B., 2007: Human Responsibility for Autonomous Agents. IEEE Intelligent Systems, 22:2, 60-61.

Stein, S., N. R. Jennings, and T. R. Payne, 2007: Provisioning Heterogeneous and Unreliable Providers for Service Workflows. Proc. 6th International Joint Conf. on Autonomous Agents and Multiagent Systems, Honolulu, HI, IFAAMAS.